Efficient Approximation of Optimal Control for Markov Games*

Markus Rabe[1], Sven Schewe[2], and Lijun Zhang[3]

[1] Universität des Saarlandes, Germany
[2] University of Liverpool, United Kingdom
[3] DTU Informatics, Technical University of Denmark, Denmark

Abstract. The success of probabilistic model checking for discrete-time Markov decision processes and continuous-time Markov chains has led to rich academic and industrial applications. The analysis of their combination in continuous-time Markov decisions processes, however, is currently restricted to toy examples. This is due to the fact that current analysis techniques for time-bounded reachability require a running time linear in the reciprocal π^{-1} of the required precision π. For the high precision usually sought (for example, six to ten digits), this simply renders these techniques infeasible. We discuss a surprising combination of discretisation and partial unravelling, which leads to memoryful near optimal schedulers that can be computed in time linear only in the square or cube root of π^{-1}. The proposed techniques also reduce the dependency on the expected number of discrete transitions within the given time bound significantly. Our techniques naturally extend to the analysis of continuous-time Markov games.

1 Introduction

Probabilistic models have been used extensively in the formal analysis of complex systems, including networked, distributed, and most recently, biological systems. While some systems can be described by probabilistic models with discrete time, for instance the random experiment of throwing a die, other system aspects, like failure behaviour, are modelled more natural with continuous-time models.

Over the past 15 years, probabilistic model checking for discrete-time Markov decision processes (MDPs) and continuous-time Markov chains (CTMCs) has been successfully applied to rich academic and industrial applications

* This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS), by the Engineering and Physical Science Research Council (EPSRC) through the grant EP/H046623/1 "Synthesis and Verification in Markov Game Structures", and by MT-LAB, a VKR Centre of Excellence.

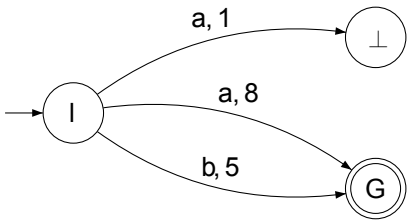

Fig. 1. This figure shows a part of a simple CTMDP. In this example, the complete probability mass is initially concentrated in location l, indicated by the incoming arrow. The goal region consists only of location G, indicated by the colour and the double line. In l, the CTMDP offers the choice between two control actions, a and b. Action a has a smaller transition rate of 5 than b, which has a higher transition rate of $1 + 8 = 9$. If the control action b is chosen, the next discrete transition will lead with a probability of $\frac{1}{9}$ to $\bot$, and with a probability of $\frac{8}{9}$ to G. Intuitively, action a leads less quickly but with higher probability (once the transition fires) to the goal location G. When time is short, a is the preferable action, while b is preferable when much time is left. We discuss efficient techniques for finding near optimal control policies for a given time bound for such CTMDPs and their extension to games.

[GMLS07,CHLS09,HMW09,BCR+09]. However, the analysis of their combination in continuous-time Markov decisions processes (CTMDPs) is currently restricted to toy examples. As for MDPs and CTMCs, the efficient approximation of the *maximum time bounded reachability probability* is of paramount significance for model checking techniques. The time-bounded reachability problem is to determine or to approximate, for a given set of goal locations G and time bound T, the maximal probability of reaching G before the deadline T. Of similar importance are the natural variations of the question, such as minimising the probability (time-bounded safety) or aiming at being in G precisely at time T (transient time-bounded reachability or safety [Mil68b]). Currently, the *efficient* analysis of CTMDPs, is limited to artificially restricted problem classes, such as schedulers without any access to time and systems with uniform transition rates [BHKH05].

For a given CTMDP with uniform transition rate λ and a time bound T, current techniques need $O(\frac{(\lambda T)^2}{\pi})$ time, where π is the required precision. In practice, the required precision is usually high—$\pi \leq 10^{-6}$—which imposes severe restrictions to the applicability of the known techniques. In this paper we propose an approach whose time complexity is merely the square, cube, or forth root of π^{-1}, and reduced dependence on the expected number of steps, resulting in $O\big(\lambda T \sqrt[k]{\frac{\lambda T}{\pi}}\big)$, where $k = 1$ essentially resembles the current techniques and $k = 2$ leads only to a marginal increase in the factor suppressed by the O notation. (For $k = 3$ and $k = 4$, there is an increasing price to pay, at least in theory.)

Our approach is based on the two existing natural ways to approximately determine optimal control and its quality for CTMDPs: Partial unravelling [BFK+09] and discretisation [NZ10].

Partial unravelling [BFK+09] would make the number of discrete transitions an explicit part of the state-space. The advantage of this technique is the fast conversion — it suffices to unravel up to a depth sub-logarithmic $o(\log \frac{1}{\pi})$ in the required precision. (For uniform CTMDPs, for example, the number of discrete events is Poisson distributed.) Unfortunately, this advantage is outweighed by the disadvantage of the intractably high costs of representing the intermediate functions with sufficient precision. Folklore therefore bans this approach as infeasible. Applying partial unravelling, the complexity for determining the optimal quality therefore has never been approached for the common scheduler classes that take time into account. But even for time abstract schedulers it is exponential in the time bound T and transition rate λ, and linear in $\frac{1}{\pi}$.

Discretisation [Mil68b] seems to be a far more practicable approach. Here, the continuous-time domain is cut into small chunks of some length ε—which one might refer to throwing an ε-net over the time—and we assume that at most one discrete transition is taken within each mesh of the ε-net. The drawback of this technique is that the precision obtained is linear in ε, while the cost is linear to its reciprocal. While currently the technique of choice [NZ10], the precision obtainable in feasible time is therefore limited.

We show that these techniques can be combined: In an interesting twist of common beliefs, we argue that, while unlimited unravelling is beyond price, unravelling once is almost free, unravelling twice is cheap, and unravelling thrice still tractable. For further unravellings, the restricting factor is that our method requires us to determine (or approximate) roots of polynomials of degree of the unravelling depth. The number of the polynomials under consideration *may* grow significantly while unravelling, but the only principle reason to restrict our attention to unravelling at most thrice is the infeasible cost of determining roots of polynomial of fourth or higher degree.

Unravelling once, twice, and thrice result in an error that is quadratic, cubic, and in the fourth power of ε, respectively. (In principle, this argument even extends to every finite number of unravellings, but with the drawbacks mentioned above.)

Our techniques naturally extend to the analysis of continuous-time Markov games and the construction of a near-optimal control strategy.

1.1 Related work

CTMDPs have been extensively studied in the control community. The analysis there has been focused on optimising expected reward [ML67,Mil68b,Mil68a,GHLPR06,BS11,Put94]. Various techniques, including *discretisation* as well as value and strategy iteration, have been exploited for the analysis.

Baier *et al.* [BHKH05] have first studied the model checking problem for CTMDPs, in which they provide an algorithm that computes time-bounded reachability probabilities in globally uniform CTMDPs. Their approach refers only to the subclass *time-abstract schedulers*, which are strictly less powerful than time-dependent ones [BHKH05,NSK09]. Recently, maximal reachability

probabilities in CTMDPs under time abstract schedulers have been studied in stochastic timed games [BF09,BFK+09]. In [NSK09], different time-abstract and time dependent schedulers are classified, together with their expressive power. It is shown that the timed schedulers depending also on the current sojourn time are the most powerful class, which agree with the one considered in this paper.

The discretisation idea has been exploited in [NZ10], and later in [CHKM10], for maximum reachability for CTMDPs and interactive Markov chains. The number of steps needed is, however, $O\left(\frac{1}{\pi}\right)$, and polynomial in λT. Therefore, these methods have limitations in their applicability to cases when high precision is required.

1.2 Organisation of the Paper

Section 2 *Preliminaries* introduces Markov games and basic notation. Section 3 *Fishing with ε-Nets* presents algorithms on normed Markov games (games with uniform transition rate 1), using different granularities of the presented techniques. We discuss how to extend our techniques to general games in Section 4 *Extensions, Generalisations, and Minor Improvements*, and describe the techniques on an example in Section 5 *Example*. Section 6 *Conclusion* concludes the paper.

2 Preliminaries

Definition 1. *A continuous-time Markov game (or simply Markov game) is defined to be a tuple $(L, L_r, L_s, \Sigma, \mathbf{R}, \mathbf{P}, \nu)$, consisting of*

- *a finite set L of locations, which is partitioned into sets of locations L_r, controlled by a* reachability *player, and L_s controlled by a* safety *player,*
- *a finite set Σ of actions,*
- *a rate matrix $\mathbf{R} : (L \times \Sigma \times L) \to \mathbb{Q}_{\geqslant 0}$,*
- *a discrete transition matrix $\mathbf{P} : (L \times \Sigma \times L) \to \mathbb{Q} \cap [0, 1]$, and*
- *an initial distribution $\nu \in Dist(L)$.*

We require that the following side-conditions hold: For all locations $l \in L$, there must be an action $a \in \Sigma$ such that $\mathbf{R}(l, a, L) := \sum_{l' \in L} \mathbf{R}(l, a, l') > 0$, which we call *enabled*. We denote the set of enabled actions in l by $\Sigma(l)$. For a location l and actions $a \in \Sigma(l)$, we require for all locations l' that $\mathbf{P}(l, a, l') = \frac{\mathbf{R}(l,a,l')}{\mathbf{R}(l,a,L)}$, and we require $\mathbf{P}(l, a, l') = 0$ for non-enabled actions. We define the *size* $|\mathcal{M}|$ of a Markov game as the number of non-null rates in the rate matrix $\mathbf{R}$.

A Markov game is called *uniform* with uniformisation rate λ, if $\mathbf{R}(l, a, L) = \lambda$ holds for all locations l and enabled actions $a \in \Sigma(l)$. We further call a Markov game *normed*, if its uniformisation rate is 1. Note that for *normed* Markov games it holds $\mathbf{R} = \mathbf{P}$.

We are particularly interested in Markov games with a single player, which are continuous-time Markov decision processes (CTMDPs). In CTMDPs all positions belong to the reachability player ($L = L_r$), or to the safety player ($L = L_s$),

depending on whether we analyse the *maximum* or *minimum* reachability probability problem.

2.1 Paths

A *timed path* σ in a Markov game $\mathcal{M}$ is a finite sequence in $L \times (\Sigma \times \mathbb{R}_{\geqslant 0} \times L)^*$:

$$l_0 \xrightarrow{a_0,t_0} l_1 \xrightarrow{a_1,t_1} \cdots \xrightarrow{a_{n-1},t_{n-1}} l_n$$

satisfying: $0 \leq t_{i-1} \leq t_i$ for all $i < n$. The t_i denote the system's time when a discrete transition from l_i to l_{i+1} takes place while the action a_i is selected. The set of all timed paths is denoted by $Paths(\mathcal{M})$, or $Paths$ if $\mathcal{M}$ is clear from the context.

2.2 Schedulers and Strategies

The non-determinism in the system needs to be resolved by a pair of strategies for the two players which together form a *scheduler* for the whole system. The power of strategies is determined by their ability to observe and distinguish paths, and thus by their domain. In this paper, we assume the most general class of strategies, the class of late *timed history-dependent* (or simply memoryful) strategies (TH) which may observe the timed path and the current time ($Paths(\mathcal{M}) \times \mathbb{R}_{\geqslant 0} \to \Sigma$). For notational convenience, we use $\mathcal{S}_r$ and $\mathcal{S}_s$ for the strategies controlling the reachability players' and the safety players' locations, respectively.

When analysing reachability objectives, we can also restrict to the even simpler class of timed positional strategies (TP) [RS10], which may observe only the current location and the total time of the system ($L \times \mathbb{R}_{\geqslant 0} \to \Sigma$).

2.3 Probability space for Markov games

We define the probability space for a Markov game $\mathcal{M}$ as the completion of the simple probability space spanned by *cylindrical schedulers* and we restrict the evolution of time to a sufficiently large interval $[0, t_{\max}]$, $t_{\max} \in \mathbb{R}_{\geq 0}$. A cylindrical scheduler assumes a partition of this interval into finitely many intervals in which it has constant decisions. A second completion on the class of cylindrical strategies then yields the full class of (measurable) TH strategies. For a pair of $(\mathcal{S}_r, \mathcal{S}_s)$ of strategies, we use $Pr_{\mathcal{S}_{r+s}}$ to denote the corresponding probability measure on paths of $\mathcal{M}$.

Note that the resulting probability space is defined on *finite* paths that have no continuation in the time interval $[0, t_{\max}]$, unlike the more common construction via the Borel σ-algebra [WJ06]. Thus, for the definition of the reachability probability (see below), it is important to consider the probability that for a finite path (or set thereof) there is no further transition after their last transition until $t_{\max}$. See [RS10] for details.

2.4 Time-bounded reachability probability

The problem we are considering is the *time-bounded reachability probability* problem. That is, given a Markov game $\mathcal{M}$, a goal region $G \subseteq L$, and a time bound $T \in \mathbb{R}_{\geq 0}$, we are interested in the set of paths that reach a location in the goal region in time:

$$reach_{\mathcal{M}}(G,T) = \Big\{ \sigma \in Paths \mid \sigma = l_0 \xrightarrow{a_0,t_0} l_1 \dots l_n \text{ with } l_n \in G \wedge t_{n-1} \leq T \\ \text{or } \exists i < n.\ l_i \in G \wedge t_{i-1} \leq T \leq t_i \Big\}.$$

Note that we only assume $t_{\max} \geq T$ for the sufficiently large time interval $[0, t_{\max}]$ referred to in the previous subsection. The definition becomes simpler when we choose $t_{\max} = T$ (in this case we simply require $l_n \in G$), but this would imply that the probability spaces for different time-bounds are formally different. (The time-bounded reachability itself would, of course, not be affected.)

We are particularly interested in *optimising* this probability and in finding the corresponding pair of strategies: $\sup_{\mathcal{S}_r \in \mathrm{TP}} \inf_{\mathcal{S}_s \in \mathrm{TP}} Pr_{\mathcal{S}_{r+s}}(reach_{\mathcal{M}}(G,T))$, which is commonly referred to as the *maximum* time-bounded reachability probability problem in the case of CTMDPs with a reachability player only.

Remark: In traditional time-bounded reachability, one would just require one of the locations on the way that is reached not later than T to be in the goal region. Technically this can be done by making the goal region absorbing. In recent literature, the term 'maximum time-bounded reachability probability' was used in in this slightly more restrictive way. We discuss this *traditional* notion in Section 4.1.

We define $f : L \times \mathbb{R}_{\geq 0} \to [0,1]$, to be the optimal probability to reach the goal region within the time limit, assuming that we start in location l and that t time units have passed already. That is, the value $f(l,t)$ is the *optimal* probability ($\sup_{\mathcal{S}_r \in \mathrm{TP}} \inf_{\mathcal{S}_s \in \mathrm{TP}} Pr_{\mathcal{S}_{r+s}}(\cdot)$) of $reach_{\mathcal{M}}(G,T)$ restricted to those paths that are in location l at time t. By definition, it holds then that $f(l,T) = 1$ if $l \in G$ and $f(l,t) = 0$ if $t \geq T$ and $l \notin G$. Optimising the vector of values $f(\cdot, 0)$ then yields the optimal strategy and its value.

2.5 Characterisation of f

The optimal function f can be characterised as the following set of differential equations [RS10]. For each $l \in L$:

1. Initial value: $f(l,T)$ equals 1 if $l \in G$, and 0 if $l \notin G$. (We do not need to define f for $t > T$.)
2. Otherwise, that is, for $t < T$, it holds:

$$-\dot{f}(l,t) = \operatorname*{opt}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{R}(l,a,l') \cdot \left(f(l',t) - f(l,t) \right), \tag{1}$$

where $\mathsf{opt} \in \{\max, \min\}$ is max for reachability player locations and min for safety player locations. We will use the opt-notation throughout this paper.

Equation (1) can be rewritten to:

$$-\dot{f}(l,t) = \operatorname*{opt}_{a \in \Sigma(l)} \left(\sum_{l' \neq l} \mathbf{R}(l,a,l') \cdot f(l',t) - \sum_{l' \neq l} \mathbf{R}(l,a,l') \cdot f(l,t) \right). \qquad (2)$$

This also provides an intuition for the fact that uniformisation does not alter the reachability probability under any strategy: the rate $\mathbf{R}(l,a,l)$ does not appear in the above reformulation.

To simplify notation, we define a matrix $\mathbf{Q}$ such that $\mathbf{Q}(l,a,l') = \mathbf{R}(l,a,l')$ if $l' \neq l$ and $\mathbf{Q}(l,a,l) = -\sum_{l' \neq l} \mathbf{R}(l,a,l')$. The characterisation above for the function f that reflects the optimal reachability probability for two rational players can then be rewritten to:

$$-\dot{f}(l,t) = \operatorname*{opt}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{Q}(l,a,l') \cdot f(l',t), \qquad (3)$$

with the same side-constraint as for Equation (1). For uniform Markov games, we simply have $\mathbf{Q}(l,a,l) = \mathbf{R}(l,a,l) - \lambda$, with $\lambda = 1$ for normed Markov games.

It is interesting to note, the above reformulation (Eqn. (3)) can be considered as a simplification of the one used in [Mil68b].

3 Fishing with ε-Nets for Normed Markov games

The standard approach to approximate optimal control can be summarised as throwing an ε-net over the time and approximating optimal control within each mesh of the net. Cost and precision depend on the number of meshes we have. More precisely, we fix the interval $[0,T]$ and mesh length ε, which then gives rise to $\lceil \frac{T}{\varepsilon} \rceil$ meshes. The global precision is then provided as the sum over the errors we allow for when moving from one mesh to the next.

To ease notation and intuition, we discuss the influence of these decisions for the case of on the example of normed Markov games. Thus, throughout the whole section, we fix a normed Markov game $\mathcal{M} = (L, L_r, L_s, \Sigma, \mathbf{R}, \mathbf{P}, \nu)$ and generalise the techniques to the full class of Markov games in section 4.3.

One gateway to Markov games and decision processes is to view them as the limit of their deterministic-time brethren [Bel57], and it is natural to invert this limit operation by considering small time intervals ε and such that the probability that more than one transition within these intervals of length ε is *small enough*.

- Our most *basic approximation scheme* is to assume that there is at most one transition within a mesh. We call it *simple ε-nets*, which shall be discussed in Subsection 3.1.
- Subsection 3.2 discusses *double ε-nets* in which there is at most two transitions within a mesh.
- Subsection 3.3 discusses *triple ε-nets* and everything beyond.

The intuition for ε-nets of level $k+1$ is that we jump to a net of level k once the first transition has occurred, but with the precise remaining time in the mesh. Based on these assumptions, we build estimators p_k for the time-bounded reachability described by f, and, of course, strategies for both players. These strategies chose the near-optimising action proposed by the estimator of the previous level in all points of time within a mesh. Thus, they to take into account in which level they currently are and are therefore memoryful (and not timed positional).

For these estimators and strategies, we discuss their precision with respect to three measures:

$\mathcal{E}(k,\tau)$ is an estimator for the difference of p and f after τ time units (to the left),

$\mathcal{E}_s(k,\tau)$ is an estimator for the difference between the time-bounded reachability for the inferred strategy and f after τ time units, and

$\mathcal{E}_p(k,\tau)$ is an estimator for the difference between the time-bounded reachability for the inferred strategy and p after τ time units.

As the names suggest, we consider the first to be slightly more important, but this is a matter of taste, and all of them have their place. We show that the step errors $\mathcal{E}(k,\varepsilon)$, $\mathcal{E}_s(k,\varepsilon)$, and $\mathcal{E}_p(k,\varepsilon)$ are all in $O(\varepsilon^{k+1})$, with very small constants. (See Subsection 4.2 for an overview.)

We exemplify a single step in Section 5.

3.1 Single ε-Nets

In *single* ε-nets, we assume that at most one transition fires within each mesh. This is closely related to assuming the schedulers to be constant within this time, resulting in a *linear estimation function* inside a mesh. Under this assumption, optimisation becomes incredibly simple:

- For the case that there is no transition, the scheduler decision does not matter.
- For the case that there is one transition, the best scheduler decision is the scheduler that optimises the expected quality after the transition is taken.

For every mesh of length ε this provides a straight-forward upper bound of the error of ε^2. But before we give this known result [NZ10,ZN10] as a warm-up, we show that for mesh-based approaches, the overall error can always be estimated by sum over the local errors.

For this, we define f as the vector valued function $f : t \mapsto \bigotimes_{l\in L} f(l,t)$ that maps each point of time to the vector of likelihoods to reach the goal region in time t for each location. On such vectors $f(t), e(t)$ (which we allow to be real valued for each location), we define the maximum norm, that is $\|f(t)-e(t)\| = \max\{|f(l,t)-e(l,t)| \mid l \in L\}$. We also refer to $e(t)$ as the *estimator*.

For a fixed mesh $[t-\varepsilon, t]$, the next notation we introduce is the vector valued function $f_x^t : \tau \mapsto \bigotimes_{l\in L} f_x^t(l,\tau)$ obtained when we use the differential equation (3), using the vector $x \in [0,1]^{|L|}$ at point t as initial values. That is, f_x^t is defined by:

1. Initial value: $f_x^t(\tau)$ equals x for $\tau = t$.
2. Otherwise, that is, for $t - \varepsilon \leq \tau < t$ and $l \in L$, it holds:

$$-\dot{f}_x^t(l,\tau) = \mathsf{opt}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{Q}(l,a,l') f_x^t(l',\tau) \tag{4}$$

Starting with an estimate $e(t)$ at the beginning of the mesh $[t-\varepsilon, t]$, the following lemma discusses the spread of error bounds, which consist of: (i) the starting error at the beginning of the mesh $\|f(t) - e(t)\|$, and (ii) the additional imprecision $\|f_{e(t)}^t(t-\varepsilon) - e(t-\varepsilon)\|$, which we refer to as the *ε-step error*.

Lemma 1. *For a given Markov game, let the vector e be an estimator of f that satisfies $\|f(t) - e(t)\| \leq \mu$ and $\|f_{e(t)}^t(t-\varepsilon) - e(t-\varepsilon)\| \leq \nu$ for some point $t \in [0,T]$. Then $\|f(t-\varepsilon) - e(t-\varepsilon)\| \leq \mu + \nu$ holds true.*

Proof: First, $\|f(t-\varepsilon) - f_{e(t)}^t(t-\varepsilon)\| \leq \mu$ can be shown by exploiting two obvious properties of the functions defined by Equation (3):

(i) When all initial values (which we assume to refer to the same time t) are in-/decreased by the same constant c then this has no effect on the derivations described in Equation (3). Hence, the resulting functions, $f_{\uparrow c}^t$, are simply in-/decreased by c for all locations and at every point of time. Note that this also changes the values in the goal region.
(ii) The values of $f_{e(t)}^t(t-\varepsilon)$ are monotonous in $e(t)$.

As $\|f(t) - e(t)\| \leq \mu$ holds true, property (ii) yields

$$\|f(t-\varepsilon) - f_{e(t)}^t(t-\varepsilon)\| \leq \|f(t-\varepsilon) - f_{\uparrow\mu}^t(t-\varepsilon)\|,$$

which implies the first claim.

$$\|f(t-\varepsilon) - e(t-\varepsilon)\| \leq \mu + \nu$$

is then implied by the triangle inequation. □

Estimator p_1 for single ε-nets Below, we shall use $p_1(t)$ to denote our linear estimator vector for single ε-nets. We start constructing p_1 by, setting $p_1(l,T) = 1$ if $l \in G$ and $p_1(l,T) = 0$ if $l \notin G$. After having constructed p_1 for the interval $[t,T]$, we expand it to the interval $[t-\varepsilon, T]$ as follows.

We first determine the optimising enabled actions for each location for $f_{p_1(t)}^t$ at time t. That is, we choose, for all $l \in L$ and all $a \in \Sigma(l)$, an action

$$a_l^t \in \arg \underset{a \in \Sigma(l)}{\mathsf{opt}} \sum_{l' \in L} \mathbf{Q}(l, a_l^t, l') \cdot p_1(l', t). \tag{5}$$

We then fix

$$c_l^t = \sum_{l' \in L} \mathbf{Q}(l, a_l^t, l') \cdot p_1(l', t) = \underset{a \in \Sigma(l)}{\mathsf{opt}} \sum_{l' \in L} \mathbf{Q}(l, a, l') \cdot p_1(l', t)$$

as the descent (that is, $-c_l^t$ as the ascent) of $p_1(l,\cdot)$ in the interval $[t-\varepsilon, t]$, which implies

$$-\dot{p}_1(l, t-\tau) = c_l^t \quad \text{and} \quad p_1(l, t-\tau) = p_1(l,t) + \tau \cdot c_l^t \tag{6}$$

for all $\tau \in [0,\varepsilon]$ and all $l \in L$.

The following lemma establishes a few properties for normed Markov games that will be used for proving our main result for single nets.

Lemma 2. *Given a normed Markov game, let $\varepsilon \leq 1$. Then, for all $\tau \in [0,\varepsilon]$ and $l \in L$, it holds:*

1. $p_1(l, t-\tau) \in [0,1]$.
2. $-\dot{f}^t_{p_1(t)}(l, t-\tau) \in [-1,1]$.

Proof: Let $\varepsilon = 1$. By (6), the value $p_1(l, t-1)$ would simply be $p_1(l,t) + \sum_{l' \in L} \mathbf{Q}(l, a_l^t, l') \cdot p_1(l', t)$. Using the rate matrix $\mathbf{R}$, we have:

$$\begin{aligned} p_1(l,t-1) &= p_1(l,t) + \sum_{l \neq l'} \mathbf{R}(l, a_l^t, l') \cdot (p_1(l',t) - p_1(l,t)) \\ &= \left(1 - \sum_{l \neq l'} \mathbf{R}(l, a_l^t, l')\right) \cdot p_1(l,t) + \sum_{l \neq l'} \mathbf{R}(l, a_l^t, l') \cdot p_1(l',t) \end{aligned}$$

For normed Markov game, we have $\mathbf{R}(l, a_l^t, L) = 1$, implying that $p_1(l, t-1) \in [0,1]$ by a simple inductive argument (starting with $p_1(l,T) \in \{0,1\}$ for all $l \in L$). For $\tau \in [0,1]$, we simply have a linear interpolation between $p_1(l, t-1)$ and $p_1(l,t)$.

Now we prove the second clause. For $\tau \in [0,\varepsilon]$, from the first part we have seen that the co-domain of p_1 is in $[0,1]^L$, and thus the same holds for $f^t_{p_1(t)}(t) = p_1(t)$. For $\tau \in [0,\varepsilon]$, as long as $f^t_{p_1(t)}(\tau)$ is in $[0,1]^L$, it holds $-f^t_{p_1(t)}(l,\tau) \leq \sum_{l' \in L} \mathbf{Q}(l,a,l') \cdot f^t_{p_1(t)}(l,\tau) \leq 1 - f^t_{p_1(t)}(l,\tau)$ for all $l \in L$ and $a \in \Sigma(l)$. These inequations hold in particular for the optimising action. As $f^t_{p_1(t)}$ adheres to the differential equations (4), it consequently cannot leave $[0,1]^L$ left of t, that is, for a $\tau \leq t$. Thus, the following the simple estimation holds, for all $a \in \Sigma(l)$:

$$\left| \sum_{l' \in L} \mathbf{R}(l,a,l') \cdot \left(f^t_{p_1(t)}(l', t-\tau) - f^t_{p_1(t)}(l', t-\tau) \right) \right| \leq \sum_{l' \in L} \left| \mathbf{R}(l,a,l') \right| = 1.$$

implying that the descent $-\dot{f}^t_{p_1(t)}(l, t-\tau)$ is in the region $[-1,1]$. □

Theorem 1. *For a normed Markov game and given $\varepsilon \leq 1$, the ε-step error for a single ε-net is bounded by $\mathcal{E}(1,\varepsilon) \leq \frac{1}{2}\varepsilon^2$.*

Proof: Lemma 2 immediately implies for all locations $l \in L$, enabled actions $a \in \Sigma(l)$, and $\tau \in [0, \varepsilon]$

$$\begin{aligned}\tau &\geq \textstyle\sum_{l' \in L} \mathbf{R}(l, a, l') \cdot \tau \\ &\geq \textstyle\sum_{l' \in L} \mathbf{R}(l, a, l') \cdot \left| f^t_{p_1(t)}(l', t-\tau) - p_1(l', t) \right| \\ &\geq \left| \textstyle\sum_{l' \in L} \mathbf{R}(l, a, l') \cdot \left(f^t_{p_1(t)}(l', t-\tau) - p_1(l', t) \right) \right|.\end{aligned}$$

This holds in particular for a^t_l (see Equation (5)) and the optimising action $a^{l-\tau}_l$ for l at time $t-\tau$ with respect to $f^t_{p_1(t)}$: $a^{t-\tau}_l \in \arg\mathsf{opt}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{Q}(l, a, l') \cdot f^t_{p_1(t)}(l', t-\tau)$. As the order of quality between a^t_l and $a^{t-\tau}_l$ is reversed when moving from t to $t-\varepsilon$, this immediately implies $\left| \dot{f}^t_{p_1(t)}(t-\tau) + c^t_l \right| \leq \tau$. To see this, consider, for example, a location $l \in L_r$ of the reachability player. We get $\sum_{l' \in L} \mathbf{R}(l, a^t_l, l') p_1(l', t) - \tau \leq \sum_{l' \in L} \mathbf{R}(l, a^t_l, l') f^t_{p_1(t)}(l', t-\tau) \leq \sum_{l' \in L} \mathbf{R}(l, a^{t-\tau}_l, l') f^t_{p_1(t)}(l', t-\tau) \leq \sum_{l' \in L} \mathbf{R}(l, a^{t-\tau}_l, l') p_1(l', t) + \tau \leq \sum_{l' \in L} \mathbf{R}(l, a^t_l, l') p_1(l', t) + \tau$.

We can therefore estimate the difference of $f^t_{p_1(t)}(t-\tau)$ and $p_1(l, t-\tau)$:

$$d(l, t-\tau) := \left| f^t_{p_1(t)}(t-\tau) - p_1(l, t-\tau) \right|$$

by observing $d(l, t) = 0$ and

$$-\dot{d}(l, t-\tau) \leq \left| \dot{f}^t_{p_1(t)}(t-\tau) - \dot{p}_1(l, t-\tau) \right| \leq \tau,$$

which implies that the accumulated error $\mathcal{E}(1, \tau)$ up to a distance τ from t is bounded by $\frac{1}{2}\tau^2$. In particular, $\mathcal{E}(1, \varepsilon) \leq \frac{1}{2}\varepsilon^2$ holds true. □

This argument can easily be extended to show similar bounds for the quality of a particular strategy. Let us partition L into two sets L_o of optimising and L_f of fixed-decision locations, in which we the action a^t_l is fixed for a given t. Let us now consider a function g

$$\begin{aligned}-\dot{g}_{L_f}(l, t) &= \textstyle\sum_{l' \in L} \mathbf{Q}(l, a^t_l, l') \cdot g_{L_f}(l', t) \text{ for all } l \in L_f, \text{ and} \\ -\dot{g}_{L_f}(l, t) &= \mathop{\mathsf{opt}}_{a \in \Sigma(l)} \textstyle\sum_{l' \in L} \mathbf{Q}(l, a, l') \cdot g_{L_f}(l', t) \text{ for all } l \in L_o.\end{aligned} \tag{7}$$

Equation (7) reflects a system where players adhere to an initially optimal action a^t_l at fixed-decision locations L_f and play optimal (under this side constraint) in optimising locations L_o. Note that the case where the set of fixed-decision locations equals the locations of the safety ($L_f = L_s$) or reachability ($L_f = L_r$) player are special cases that are equivalent to the problem of optimising the reachability probability for a CTMDP. These special cases describe a strategy for the respective player, namely the strategy to follow the initially optimal decision a^t_l throughout the complete ε-mesh $[t-\varepsilon, t]$.

From a technical perspective, these equations equal the equations that we get when changing the Markov game itself by fixing the decisions of all players in L_f. For this adjusted normed Markov game, we have the same estimator p_1 on the interval $[t-\varepsilon, t]$ when starting with the same initial values $p(t)$. We therefore obtain similar bounds by a simple corollary from Theorem 1:

Corollary 1. *For $\varepsilon \leq 1$, the ε-step error for a normed Markov game in a single ε-net with any subset of initially fixed-decision locations L_f is bounded by $\mathcal{E}_p(1,\varepsilon) \leq \frac{1}{2}\varepsilon^2$.*

Obviously, the absolute value of the difference between $g(t-\tau)$ and $f^t_{p_1(t)}(t-\tau)$ can be estimated by $|g(t-\tau) - p_1(t-\tau)| + |f^t_{p_1(t)}(t-\tau) - p_1(t-\tau)|$, which immediately provides us with a $\mathcal{E}_s(1,\varepsilon) \leq \mathcal{E}(1,\varepsilon) + \mathcal{E}_p(1,\varepsilon) \leq \varepsilon^2$ bound for this difference.

Corollary 2. *For a normed Markov game and any pair of fixed-decision locations $L_1, L_2 \subseteq L$, the ε-step error $\|g_{L_1}(t-\varepsilon) - g_{L_2}(t-\varepsilon)\|$ is bounded by $\mathcal{E}_s(1,\varepsilon) \leq \varepsilon^2$.*

In particular, the above corollary implies $\left\| f^t_{p_1(t)}(t-\varepsilon) - g_{L_r}(t-\varepsilon) \right\| \leq \varepsilon^2$ and $\left\| f^t_{p_1(t)}(t-\varepsilon) - g_{L_s}(t-\varepsilon) \right\| \leq \varepsilon^2$, and hence provides the sought bound for the error of the approximate strategies of the safety and reachability player, respectively.

Corollary 3. *When using single ε-nets, the overall error for normed Markov games with time bound T is bounded by εT, both for the estimation of f and for the approximation of the value for a computed strategy.*

As expected, this technique based on simple ε-nets provides us with costs and guarantees comparable to those established by Neuhäußer and Zhang [NZ10]:

Theorem 2. *For a normed Markov game $\mathcal{M}$ of size $|\mathcal{M}|$, we can compute a π-optimal strategy and determine the quality of $\mathcal{M}$ up to precision π in time $O(|\mathcal{M}| \cdot T \cdot \frac{T}{\pi})$.*

Proof: To guarantee an overall precision π, we can choose $\varepsilon = \lceil \frac{\pi}{T} \rceil$, resulting in $\frac{T}{\varepsilon} \approx \frac{T^2}{\pi}$ many steps. The cost of each step is dominated by the cost for computing and comparing for each location the chances of winning when a particular action is taken. This cost[4] can be estimated by $|\mathcal{M}|$. □

[4] It is always a matter of taste if one wants to be precise and consider the contribution of numerical precision. A suitable precision can be obtained by using numbers of size $O(\log \frac{1}{\pi})$, which leads to $O(\log \frac{1}{\pi})$ cost for the multiplications. We have dropped this cost to be consistent with most other work.

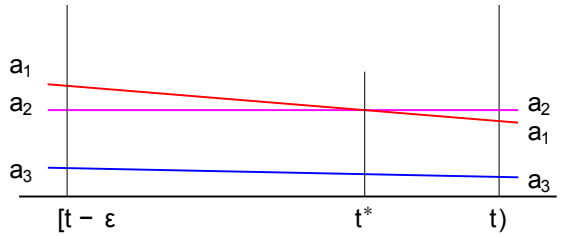

Fig. 2. This figure illustrates (linear) quality estimates $z(a) := \sum_{l' \in L} \mathbf{R}(l, a, l') \cdot (p_1(l', \tau) - p_1(l, \tau))$ for three control actions a_1, a_2, and a_3. Assuming we are considering a location of the reachability player, the function $-\dot{p}_2(l, \tau)$ agrees with the quality of a_1 for the interval $[t - \varepsilon, t^*)$, followed by a_2 for the interval $[t^*, t)$. We refer to Section 5 for a detailed example of computation for both $\dot{p}_2$ and p_2.

3.2 Double ε-Nets

Single ε-nets provide an easy gateway to recent results for CTMDPs, and their simple extensions to games. In this subsection we show the surprising result that we can improve the precision significantly for very low costs by increasing the number of discrete events we allow for within each mesh. The increased precision allows us to enlarge the width of a mesh. We can, therefore, traverse the time axis considerably faster.

The idea of double ε-nets is to use two different layers of behaviour to describe the near optimal scheduler. The first layer is used until the first transition happens, while the second is used for any further behaviour. Thus, the scheduler we would obtain is memoryful: it distinguishes the situations where the first discrete transition has and has not occurred. For the first layer of behaviour we simply use single ε-nets; we fix an action for the complete ε-interval. For the behaviour after the first jump (i.e., the second layer) we additionally consider the possibility that for time points $t - \tau \in (t - \varepsilon, t]$ the near optimal action (according to the estimator p_1) changes.

This idea leads to the following estimator p_2 for double ε-Nets. For the mesh $[t - \varepsilon, t]$, p_2, initialised at time t, is defined as:

$$-\dot{p}_2(l, \tau) = \operatorname*{opt}_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{R}(l, a, l') \cdot (p_1(l', \tau) - p_1(l, \tau)) \,. \tag{8}$$

From the previous subsection, we have seen that the optimal strategy for an ε-interval can be approximated with precision ε^2 by a constant strategy, for example a_l^t for location l under consideration. For each $a \in \Sigma(l)$, $\sum_{l' \in L} \mathbf{R}(l, a, l') \cdot (p_1(l', \tau) - p_1(l, \tau))$ is simply a linear function. Two lines intersect at most once. Thus, $-\dot{p}_2(l, \tau)$ is piecewise linear, which is the fringe of a segment of a planar polyhedron. For the small interval we look at, the optimising decision will often be constant and scarcely change more than once. In Figure 2 we illustrate the case when the near optimal action changes during an interval.

Since $\dot{p}_2(l, \tau)$ is piecewise linear, $p_2(l, \tau)$ is therefore piecewise quadratic. We show below that the resulting piecewise quadratic estimation improves the error bound of a ε-mesh to $\frac{1}{3}\varepsilon^3$.

Lemma 3. For a normed Markov game and $\varepsilon \leq 1$, the ε-step error under double ε-nets is $\mathcal{E}(2,\varepsilon) \leq \frac{1}{3}\varepsilon^3$.

Proof: The proof runs similar to the proof of Theorem 1; all we need to do is to adjust the differential equations in the proof of Theorem 1, replacing the coarse approximation with respect to the constant by a more precise estimator.

Using p_1, the results from Theorem 1 provide $\|p_1(\tau) - f^t_{p_1(t)}(\tau)\| \leq \frac{1}{2}(t-\tau)^2$ for all $\tau \in [t-\varepsilon, t]$.

Now we switch to the estimator p_2 defined by Equation 8. For each action $a \in \Sigma(l)$, the following holds: $\sum_{l' \in L} \mathbf{R}(l,a,l') \cdot (p_1(l',\tau) - p_1(l,\tau))$ is always in within a $(t-\tau)^2$ margin from $\sum_{l' \in L} \mathbf{R}(l,a,l') \cdot \left(f^t_{p_1(t)}(l',\tau) - f^t_{p_1(t)}(l,\tau)\right)$ for normed Markov games. Using triangle inequations, we can again lift this to $\|\dot{p}_1(\tau) - \dot{f}^t_{p_1(t)}(\tau)\| \leq (t-\tau)^2$, and infer $\|p_2(\tau) - f^t_{p_1(t)}(\tau)\| \leq \frac{1}{3}(t-\tau)^3$. □

Like for single ε-nets, we can extend this to a constructive version, paying only with a slightly higher constant factor.

Lemma 4. For $\varepsilon \leq 1$, we can provide a memoryful strategy whose ε-step error for a normed Markov game is $\mathcal{E}_s(2,\varepsilon) \leq \frac{1}{2}\varepsilon^3$.

Proof: To estimate the error incurred in a single step, we consider the point in time where the *first* discrete transition occurs in the ε interval under consideration. (Note that the distribution of this time is independent from the strategy.)

Provided that no discrete transition occurs in the interval, the quality of the outcome does not depend on the chosen strategy—we simply stay in the current location. If a transition occurs with τ time units left, we jump to the next lower level of ε-nets, but with τ (not ε) time units left. That is, we enter a simple τ-net. The error occurring *provided* the first discrete transition $\tau \in [0,\varepsilon]$ time units prior to the end of the interval is therefore $\mathcal{E}(1,\tau)$, simply because this is the bound for the error (and in particular for the expected error in randomised transitions) at the target location.

We therefore get an immediate estimation for the error that occurs in every step for double ε-nets:

$$\begin{aligned}\mathcal{E}_s(2,\varepsilon) &\leq \int_0^\varepsilon e^{\tau-\varepsilon}\big(\mathcal{E}_s(1,\tau) + \mathcal{E}_p(1,\tau)\big)d\tau \\ &\leq \int_0^\varepsilon \big(\mathcal{E}_s(1,\tau) + \mathcal{E}_p(1,\tau)\big)d\tau = \int_0^\varepsilon 1\frac{1}{2}\tau^2 d\tau = \frac{1}{2}\varepsilon^3.\end{aligned}$$

□

As a simple corollary of the previous lemmata, the precision with which p_2 predicts the value of the respective strategy is bounded by the sum of these deviations.

Corollary 4. For $\varepsilon \leq 1$, the ε-step error in a normed Markov game, under double ε-net, for the prediction of the quality of the inferred strategy is $\leq \mathcal{E}_p(2,\varepsilon) \leq \frac{5}{6}\varepsilon^3$.

The evaluation of the estimator p_2 is not expensive: p_1 is linear, $\dot{p}_2$ piecewise linear, and p_2 therefore piecewise quadratic.

This raises our estimation of step cost slightly: we can only estimate the (average) number of changes for each location to be linear. This estimated number of switching points leads to an estimated time for the sorting in the algorithm described above each step and each location of $|\Sigma| \log |\Sigma|$. (Note that one of these Σ's later is included in $|\mathcal{M}|$.)

In the following, we give an algorithm that computes these linear functions describing the quality.

Algorithm:

We do the following steps for every position l, for mesh $[t - \varepsilon, t]$.

1. Compute the quality of the decisions when no time is left, and when ε time units are left, storing the respective optimal decisions a_l^0 (for t) and a_l^ε (for $t - \varepsilon$) on the way. (0-Quality[5] & ε-Quality. Cost: $O(|\Sigma(l)|)$.)
2. If $a_l^0 = a_l^\varepsilon$ then we chose this decision for the complete interval.
   Otherwise, disregard all decisions for which the 0-quality is inferior to the 0-quality of a_l^ε or for which the 0-quality is inferior to the 0-quality of a_l^ε. (Preprocessing. Cost: $O(|\Sigma(l)|)$.)
3. Order the quality of the decisions when no time is left. (0-Order. Cost: $|\Sigma| \log(|\Sigma|)$.)
4. We start with the top element of the 0-order, and choose it initially for the complete interval. We store the decision, its quality, and the left and right borders in a stack.
5. We then successively consider the next action a of the 0-order, and compare its ε-quality with the ε-quality of the element at the top of the stack.
   If the ε-quality of a does not improve over ε-quality of the top of the stack, we go to the next action in the 0-order (Step 5). Otherwise,
   (a) Check where it intersects with the current top element.
   If they intersect on or right of the of the right border of the current top element then we pop this element and go back to Step 5a.
   If they intersect left of the right border of the current top element then we set the left border of the current top element to the intersection, and push the current element with $-\varepsilon$ as left border, and this intersection point as right border. We then proceed with the next element in the 0-Order (Step 5).

This algorithm's complexity is $|\mathcal{M}| \cdot \log(|\Sigma|)$ for a normed Markov game. For each ε-step, the cost to compute an update consists of two parts:

- Computing the (linear) quality functions for all actions in all locations.
- Constructing the piecewise linear function for each location of these near optimal solutions, as described in the algorithm above.

[5] The 0-Quality and a_l^t have already been computed for the underlying single ε-net.

The first step is essentially what we do in case of single nets as well, and it has the same cost known from ordinary discretisation: $O(|\mathcal{M}|)$. For the second step, we need to sort the $|\Sigma|$ many functions (for every location) and then sweep through the obtained order. As the sorting only depends on the quality functions computed previously, we can sort with a complexity of $O(|L| \cdot |\Sigma| \log(|\Sigma|))$. As $L \cdot |\Sigma|$ is dominated by $|\mathcal{M}|$, the complexity of the algorithm is $O(|\mathcal{M}| \cdot \log(|\Sigma|))$.

Remark 1. While we consider the $\log |\Sigma|$ part of the sorting algorithm, we argue that it will not occur in practice. If the number of successors is small, this factor is irrelevant. But even if the factor is high, we believe that our preprocessing step will usually reduce the actions under consideration significantly.

Note that the $\log |\Sigma|$ factor already vanishes if only the $\frac{1}{\log(|\Sigma|)}$ part of the successors have this property *on average*. It is our believe that the likelihood of this is very low.

Lemma 5. For a normed Markov game $\mathcal{M}$ the cost of an approximative evaluation of an ε mesh (step-costs) of a double ε-net is in $O(|\mathcal{M}| + |L| \cdot |\Sigma| \cdot \log |\Sigma|)$.

To derive a precision π with a double ε-net by Lemma 1, we choose $\varepsilon \approx \sqrt{\frac{\pi}{T}}$, resulting in $\frac{T}{\varepsilon} \approx \frac{T^{1.5}}{\sqrt{\pi}}$ steps.

Corollary 5. For a normed Markov game $\mathcal{M}$ we can approximate the time-bounded reachability, construct π optimal memoryful strategies for both players, and determine the quality of these strategies with precision π in time $O(|\mathcal{M}| \cdot T \cdot \sqrt{\frac{T}{\pi}} + |L| \cdot T \cdot \sqrt{\frac{T}{\pi}} \cdot |\Sigma| \log |\Sigma|)$.

There are, in our view, good reasons to disregard the extra $\log |\Sigma|$ complexity: The number of *relevant* actions in the sense that they have effect within a specific ε-mesh (that is, the number of points where OPT 'switches' is decision, like t^* in Figure 2), is bound to be tiny in practice. As long as the number of relevant actions (or: switching points) does not exceed $\frac{|\Sigma|}{\log |\Sigma|}$ in the average, the additional cost $\log |\Sigma|$ vanishes completely, even if $|\mathcal{M}|$ has only deterministic actions. Moreover, in a Markov game with many randomised decisions (such as the control action b in Figure 1) with a high ($\geq \log |\Sigma|$) expected potential successors per control action, the logarithmic factor is always masked.

3.3 Triple ε-Nets and Beyond

Taking the huge advantage we could yield by going from one to two discrete steps into account, the question whether we could (and should) go further begs to be asked. And this is indeed the case: If we revisit the proof of Lemma 3 and its extension to Lemma 4, then it is quite apparent that the guarantees do not depend on the fact that the estimator p_1 is piecewise linear, it merely depends on its precision.

Let us inductively define estimators p_k for epsilon nets of level k following the same pattern as for double ε nets:

$$-\dot{p}_k(l,\tau) = \mathop{\mathsf{opt}}_{a\,\in\,\Sigma(l)} \sum_{l'\in L} \mathbf{R}(l,a,l')\cdot(p_{k-1}(l',\tau) - p_{k-1}(l,\tau)) \tag{9}$$

with initial values $p_k(t) = p_{k-1}(t)$. Then we can repeat the arguments from the previous subsection on each level and obtain:

Theorem 3. $\mathcal{E}(k,\varepsilon) \le \frac{2\varepsilon}{k+1}\mathcal{E}(k-1,\varepsilon)$, $\mathcal{E}_s(k,\varepsilon) \le \frac{\varepsilon}{k+1}\big(\mathcal{E}(k-1,\varepsilon)+\mathcal{E}_p(k-1,\varepsilon)\big)$, $\mathcal{E}_p(k,\varepsilon) \le \mathcal{E}(k,\varepsilon) + \mathcal{E}_p(k,\varepsilon)$.

Proof: The argument is simply the inductive version of the arguments from Lemmata 3 and 4 and Corollary 4, where single or double nets and their estimators and strategies are replaced by nets of level $k-1$ or k, respectively, and their estimators and strategies. □

The resulting precisions for nets of level k are:

k	1	2	3	4	...
$\mathcal{E}(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{3}\varepsilon^3$	$\frac{1}{6}\varepsilon^4$	$\frac{1}{15}\varepsilon^5$	...
$\mathcal{E}_s(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{2}\varepsilon^3$	$\frac{1}{3}\varepsilon^4$	$\frac{1}{6}\varepsilon^5$	...
$\mathcal{E}_p(k,\varepsilon)$	ε^2	$\frac{5}{6}\varepsilon^3$	$\frac{1}{2}\varepsilon^4$	$\frac{7}{30}\varepsilon^5$	...

Note that the constant factors in front of the ε^{k+1} are clearly decreasing quickly (super-exponentially in k). It is also apparent that we can infer the required numbers of steps in an equally simple fashion as for single an double ε-nets:

Lemma 6. To derive a precision π with an ε-net of level k, we choose $\varepsilon \approx \sqrt[k]{\frac{\pi}{T}}$, resulting in $\frac{T}{\varepsilon} \approx \frac{T^{1\,\frac{1}{k}}}{\sqrt[k]{\pi}}$ steps.

Proof: For such an ε, the step error is by Theorem 3 is approximately $\left(\frac{\pi}{T}\right)^{1\,\frac{1}{k}}$ and the claimed number of steps is apparent. (Note that the $\approx$ hides a factor *bigger* than one.) With Lemma 1, it follows that the overall error is bounded by π. □

While the estimation so for suggested good news, the flaw of the method is the price tag attached to the individual steps. To continue with the good news for a moment, we first constitute that the individual estimators p_k are simple:

Lemma 7. *p_k is piecewise polynomial with degree $\le k$.*

Proof: The definition of p_k invites a simple inductive argument: We have already demonstrated the claim for p_1 (which is linear) and p_2. If we have established that $k-1$ is piecewise polynomial of degree $\le k-1$, then so are $\sum_{l'\in L}\mathbf{Q}(l,a,l')p_{k-1}(l',\cdot)$, $\mathsf{opt}_{a\in\Sigma(l)}\sum_{l'\in L}\mathbf{Q}(l,a,l')p_{k-1}(l',\cdot)$, and hence $\dot{p}_k$. Consequently, p_k is piecewise polynomial with degree $\le k$. □

A practical problem with the argument is that the property of being piecewise polynomial is preserved, but not the number of pieces. The number of pieces is

resistant to a realistic estimation: It is our belief that the number of pieces is usually small, because in a small interval the optimal decisions do not tend to change very often.

The worst case occurs if every location can be reached from every location through every action in a single step. (Which is also not a very realistic assumption.) For a rough estimation, we start in every level with the number of pieces from the previous level. For the comparison under the OPT function this means, for every location $l \in L$, up to $O(|\Sigma(l)|^2)$ times as many pieces compared to the previous level, and hence up to $O(|L| \cdot |\Sigma|^2)$ pieces overall.

For single nets, we started with one piece, and for double nets we had an argument that restricted the growth to $|\Sigma(l)|$ pieces for each location. For nets of higher order, however, we can only offer only the course estimation that the number of changes is in $O(|L| \cdot |\Sigma|^3)$, $O(|L|^2 \cdot |\Sigma|^5)$, ... for triple nets, quadruple nets, and so forth. While we do not believe that this number is anywhere close to realistic, we have to acknowledge that both the search depth and the length ε of a mesh grow with the level of the nets. The likelihood of getting many (though probably not in the order of the worst case estimation) pieces is bound to grow, too.

Lemma 8. If we can estimate the number of polynomial pieces in p_{k-1} in an ε-mesh by c, then the number of polynomial pieces for p_k is in this ε-mesh is $\frac{1}{2} \cdot c \cdot k \cdot |L| \cdot |\Sigma|^2$.

Proof: If we consider all actions compared by OPT for a particular location $l \in L$, we may have $|\Sigma(l)|$ functions consisting of c pieces (with similar borders). For each piece, we have $\frac{1}{2}|\Sigma(l)|\big(|\Sigma(l)| - 1\big)$ comparisons between different letters, where we compare two polynomials of degree $\leq k$. To estimate the changing points, we simply estimate the number of roots of the the difference of these polynomials (disregarding equal polynomials) for each pair of letters. The numbers of roots for each pair is bounded by k, which allows for estimating the number of pieces for the location l by $\frac{1}{2}|\Sigma(l)|^2$.

This immediately provides the claimed bound on the pieces for all locations in the complete ε-mesh. □

There are several obvious improvements for this bound. In particular, one gets much better bounds for *given* Markov games: For goal location there is always only one piece, no matter on which level. For all other locations l, one can start with the estimation $\mathit{pieces}(2, l) \leq |\Sigma(l)|$ for double nets.

Based on this, one can get the number of swapping points $\mathit{swaps}(k, l)$ of a location on level k (where $\mathit{swaps}(k, l) = \mathit{pieces}(k, l) - 1$) by

$$\mathit{swaps}(k+1, l) \leq k \cdot \sum_{a,b \in \Sigma(l), a \neq b} 1 + \sum_{l' \in l \xrightarrow{a} \cup l \xrightarrow{b}} \mathit{swaps}(k, l'),$$

where $l \xrightarrow{a}$ denotes the locations reachable in a single discrete transition with action a from l. For our example net, for example this reduces the number of pieces per location to

level	1	2	3	4	...
l_S	1	2	4	12	...
l_R	1	2	6	27	...
l	1	1	1	1	...
$\perp$	1	1	1	1	...
G	1	1	1	1	...

which we believe is still a coarse estimation.

However, there is a very practical problem as well: To find (or approximate with sufficient precision) p_k, we have to determine (or approximate with sufficient precision) the roots of polynomials of degree up to $k-1$. We also have to store and sort the potential switching points, in particular the roots of differences of polynomials within the interval borders, and sort them.

While storing and sorting is simple, we can only cheaply determine the roots of quadratic functions, and approximate the roots of cubic functions[6]. This effectively restricts the applicability of the proposed technique to quadruple nets, because the for ε-nets of higher order we would have to approximate polynomials of higher degree, which is computationally expensive.

For triple and quadruple ε-nets, the sorting of the potential switching points is the dominating cost factor in the estimation of the running time of our algorithm, just as it was for double ε-nets.

Corollary 6. For a a normed Markov game with $|L|$ locations, a goal region G and a time bound T we can construct π optimal memoryful strategies for both players and determine the quality of these strategies with precision π in time $O(|L|^2 \cdot \sqrt[3]{\frac{T}{\pi}} \cdot T \cdot |\Sigma|^3 \log |\Sigma|)$ when using quadruple ε-nets.

For a a normed Markov game with $|L|$ locations, a goal region G and a time bound T we can construct π optimal memoryful strategies for both players and determine the quality of these strategies with precision π in time $O(|L|^3 \cdot \sqrt[4]{\frac{T}{\pi}} \cdot T \cdot |\Sigma|^5 \log |\Sigma|)$ when using quadruple ε-nets.

It is open if the extensions to triple and quadruple ε-nets will only be of theoretical interest, or if they make it into practice. One should note that the extra cost can be avoided when we know that the decision in a location will remain stable during the complete ε-mesh of a triple or quadruple ε-net. This invites the development of combined techniques, where we use quadruple and triple ε-nets most of the time, but switch to double ε-nets with finer meshes in the rare event where really many switches occur.

4 Extensions, Generalisations, and Minor Improvements

In this section, we describe the trivial extension of our techniques to traditional reachability (or combinations of transient and traditional time-bounded reacha-

[6] For cubic functions, one root can be approximated efficiently. Knowing an approximate root one can perform polynomial division and discard of the sufficiently small remainder.The resulting quadratic functions can again be treated easily.

bility), minor improvements that can be gained for the ε_p, and the generalisation to general games.

4.1 Traditional Bounded-Reachability

In traditional time-bounded reachability, we can exploit that for $l \in G$ this implies $f_x^t(l,t) = x(l)$ and $-\dot{f}_x^t(l,\tau) = 0$, since these locations are absorbing. For the occurring estimators, always get $f_x^t(l,\tau) = 1$.

This can be used to improve the bound of $\mathcal{E}(1,\varepsilon)$ to $\frac{1}{2}\varepsilon^2$, because it implies that the values of f and (for $\varepsilon \leq 1$) p_1 are location-wise monotonously decreasing.

Theorem 4. *For a normed Markov game and any pair of fixed-decision locations $L_1, L_2 \subseteq L$, the ε-step error $\|g_{L_1}(t-\varepsilon) - g_{L_2}(t-\varepsilon)\|$ is bounded by $\mathcal{E}_s(1,\varepsilon) \leq \frac{1}{2}\varepsilon^2$.*

Proof: In order to prove this, we first acknowledge that turning more locations of the reachability (safety) player into fixed-decision locations decreases (increases) the value of g (not necessarily strictly) on every point left of t. It thus suffices to estimate $g_{L_r}(t-\tau) - g_{L_s}(t-\tau)$ for all $\tau \in [0,\varepsilon]$.

The second observation is that the values of $g_{L'}(l,\cdot)$ is falling monotonously left of t, and that the value of $-g_{L'}(l,t-\tau)$ is in $[0,1]$ for all $L' \subseteq L \ni l$ and all $\tau \geq 0$. Let

$$a_l^{t-\tau} = \arg\max_{a \in \Sigma(l)} \sum_{l' \in L} \mathbf{R}(l,a,l') g_{L_r}(l', t-\tau).$$

Then we have

$$\begin{aligned}
\textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^t, l') g_{L_s}(l', t-\tau) &\leq \textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^t, l') g_{L_r}(l', t-\tau) \\
&\leq \textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^{t-\tau}, l') g_{L_r}(l', t-\tau) \\
&\leq \textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^{t-\tau}, l') g_{L_r}(l', t) + \tau \\
&\leq \textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^{t}, l') g_{L_r}(l', t) + \tau \\
&= \textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^{t}, l') g_{L_s}(l', t) + \tau \\
&\leq \textstyle\sum_{l' \in L} \mathbf{R}(l, a_l^{t}, l') g_{L_s}(l', t-\tau)) + \tau,
\end{aligned}$$

and in particular

$$\sum_{l' \in L} \mathbf{R}(l, a_l^{t-\tau}, l') g_{L_r}(l', t-\tau) - \sum_{l' \in L} \mathbf{R}(l, a_l^{t}, l') g_{L_s}(l', t-\tau) \in [0, \tau]$$

Using this, we can estimate for locations $l \in L_r$:

$$
\begin{aligned}
&|\dot{g}_{L_r}(l,t-\tau) - \dot{g}_{L_s}(l,t-\tau)| \\
&= \Big| \big(\max_{a\in\Sigma(l)} \textstyle\sum_{l'\in L} \mathbf{R}(l,a,l') g_{L_r}(l',t-\tau) - g_{L_r}(l,t-\tau) \big) \\
&\qquad - \big(\textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^t,l') g_{L_s}(l',t-\tau) - g_{L_s}(l,t-\tau) \big) \Big| \\
&= \Big| \big(\max_{a\in\Sigma(l)} \textstyle\sum_{l'\in L} \mathbf{R}(l,a,l') g_{L_r}(l',t-\tau) - \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^t,l') g_{L_s}(l',t-\tau) \big) \\
&\qquad - \big(g_{L_r}(l,t-\tau) - g_{L_s}(l,t-\tau) \big) \Big| \\
&\leq \max \Big\{ \max_{a\in\Sigma(l)} \textstyle\sum_{l'\in L} \mathbf{R}(l,a,l') g_{L_r}(l',t-\tau) - \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^t,l') g_{L_s}(l',t-\tau), \\
&\qquad\qquad g_{L_r}(l,t-\tau) - g_{L_s}(l,t-\tau) \Big\} \\
&\leq \tau.
\end{aligned}
$$

Similarly, let

$$
a_l^{t-\tau} = \arg\min_{a\in\Sigma(l)} \sum_{l'\in L} \mathbf{R}(l,a,l') g_{L_s}(l',t-\tau).
$$

for all locations $l \in L_s$ of the safety player. Then we have

$$
\begin{aligned}
\textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t-\tau},l') g_{L_s}(l',t-\tau) &\leq \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t-\tau},l') g_{L_r}(l',t-\tau) \\
&\leq \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t},l') g_{L_r}(l',t-\tau) \\
&\leq \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t},l') g_{L_r}(l',t) + \tau \\
&= \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t},l') g_{L_s}(l',t) + \tau \\
&\leq \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t-\tau},l') g_{L_s}(l',t) + \tau \\
&\leq \textstyle\sum_{l'\in L} \mathbf{R}(l,a_l^{t-\tau},l') g_{L_s}(l',t-\tau)) + \tau,
\end{aligned}
$$

which again provides us with

$$
\sum_{l'\in L} \mathbf{R}(l,a_l^{t-\tau},l') g_{L_r}(l',t-\tau) - \sum_{l'\in L} \mathbf{R}(l,a_l^{t},l') g_{L_s}(l',t-\tau) \in [0,\tau],
$$

allowing for the same approximation.

Hence, $\|\dot{g}_{L_r}(t-\tau) - \dot{g}_{L_s}(t-\tau)\| \leq \tau$, which implies

$$
\|g_{L_r}(t-\tau) - g_{L_s}(t-\tau)\| \leq \frac{1}{2}\tau^2
$$

by integration. □

As these better bounds would enter into all estimations of $\mathcal{E}_p(k,\varepsilon)$ and $\mathcal{E}_p(k,\varepsilon)$, these values are improved as well, resulting in the following estimations:

k	1	2	3	4	...
$\mathcal{E}(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{3}\varepsilon^3$	$\frac{1}{6}\varepsilon^4$	$\frac{1}{15}\varepsilon^5$	...
$\mathcal{E}_s(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{3}\varepsilon^3$	$\frac{1}{4}\varepsilon^4$	$\frac{2}{15}\varepsilon^5$	...
$\mathcal{E}_p(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{2}{3}\varepsilon^3$	$\frac{5}{12}\varepsilon^4$	$\frac{1}{5}\varepsilon^5$	...

4.2 Added Precision for Free

The precisions discussed so far are easy to improve $\mathcal{E}_p(k,\varepsilon)$: The error for $\mathcal{E}_s(k,\varepsilon)$ for the strategies for the safety and reachability player are single sided. Using this single sidedness provides for an improvement of $\mathcal{E}(k,\varepsilon)$:

Theorem 5. *For the estimator $\overline{p}_k(t-\tau) = p_k(t-\tau) - \frac{1}{2} \uparrow \mathcal{E}_s(k,\tau)$ for the strategy of the safety and $\overline{p}_k(t-\tau) = p_k(t-\tau) - \frac{1}{2} \uparrow \mathcal{E}_s(k,\tau)$ for the strategy of the reachability player, respectively, the adjusted estimated of the step error for estimating the quality of the prediction for the time-bounded reachability of the resulting strategy improves to $\overline{\mathcal{E}}_p(k,\varepsilon) \leq \mathcal{E}(k,\varepsilon) + \frac{1}{2}\mathcal{E}_s(k,\varepsilon)$.*

Proof: The quality of obtained by following any fixed strategy for the safety (reachability) player naturally yields a lower (higher) time-bounded reachability probability compared to this player playing optimal. (Where lower / higher does not necessarily refer to strictly lower / higher.)

Hence, we know that the difference between the time-bounded reachability for the co-optimal strategies and an estimated start value x, f_x^t, and for the fixed strategy at time $t-\tau$ are bounded by $\mathcal{E}_s(k,\tau)$, and we know that the value for the fixed strategy is lower in every point when fixing the strategy of the safety player. Consequently, we know that the difference between the co-optimal strategy and the value for a fixed strategy of the safety player (with optimal response) is within a $\frac{1}{2}\mathcal{E}_s(k,\tau)$ *around* $f_x^t - \frac{1}{2} \uparrow \mathcal{E}_s(k,\tau)$.

$f_x^t - \frac{1}{2} \uparrow \mathcal{E}_s(k,\tau)$, in turn, is within an $\mathcal{E}(k,\tau)$ margin of $\overline{p}_k(t-\tau) = p_k(t-\tau) - \frac{1}{2} \uparrow \mathcal{E}_s(k,\tau)$ by Theorems 1 and 3. Using triangle inequations we obtain the claimed result for the strategy of the safety player.

The proof for the reachability player runs accordingly. □

The reason that we did not introduce this improvement in the respective subsection is that using this once destroys the single sidedness of the error, and the result is best if we do it only in the final step. It also increases the degree of the polynomials involved. However, this never poses a problem when used on the top level because it suffices to adjust the value at the fringes of the interval, Not leading to any extra cost to speak of. ($O(|L| \cdot \log k)$ arithmetic operation in every step.)

Note that it would not impose a principle problem to use this in an earlier step instead, as long as we use this adjustment for all locations (including goal locations), disregarding the fact that this might cause them to leave the $[0,1]$ boundary. If we do make corrections there, the *differences* can reach a higher degree, which could, in principle, lead to problems similar to the ones we only face in the nets of the next level.

The following table shows the minor improvement this estimator for the quality of the resulting strategy yields for transient time-bounded reachability:

k	1	2	3	4	...
$\mathcal{E}(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{3}\varepsilon^3$	$\frac{1}{6}\varepsilon^4$	$\frac{1}{15}\varepsilon^5$	...
$\mathcal{E}_s(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{2}\varepsilon^3$	$\frac{1}{3}\varepsilon^4$	$\frac{1}{6}\varepsilon^5$	...
$\mathcal{E}_p(k,\varepsilon)$	ε^2	$\frac{5}{6}\varepsilon^3$	$\frac{1}{2}\varepsilon^4$	$\frac{7}{30}\varepsilon^5$	...
$\overline{\mathcal{E}}_p(k,\varepsilon)$	$\frac{3}{4}\varepsilon^2$	$\frac{7}{12}\varepsilon^3$	$\frac{1}{3}\varepsilon^4$	$\frac{3}{20}\varepsilon^5$	...

The following table shows the minor improvement this estimator for the quality of the resulting strategy yields for traditional time-bounded reachability:

k	1	2	3	4	...
$\mathcal{E}(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{3}\varepsilon^3$	$\frac{1}{6}\varepsilon^4$	$\frac{1}{15}\varepsilon^5$	...
$\mathcal{E}_s(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{3}\varepsilon^3$	$\frac{1}{4}\varepsilon^4$	$\frac{2}{15}\varepsilon^5$	...
$\mathcal{E}_p(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{2}{3}\varepsilon^3$	$\frac{5}{12}\varepsilon^4$	$\frac{1}{5}\varepsilon^5$	...
$\overline{\mathcal{E}}_p(k,\varepsilon)$	$\frac{1}{2}\varepsilon^2$	$\frac{1}{2}\varepsilon^3$	$\frac{7}{24}\varepsilon^4$	$\frac{2}{15}\varepsilon^5$	...

4.3 From Normed to General Markov Games

The use of *normed* Markov games in the previous section is for convenience only. The use of general games would force to use the transition rate as a factor at different points, which needlessly impeded readability. We start with illustrating how to generalise our techniques for Markov chains, and then show that it can be carried over to Markov games.

Markov Chains A continuous-time Markov chain (CTMC) is a Markov game with a singleton set of actions $\{a\}$, thus it is connected to a unique stochastic process. Let $\pi(t)$ denote the transient probability vector at time t. The transient probability is given by [Ste94]: $\pi(t) = \nu e^{\mathbf{Q}t}$. The analytical solution can be reformulated as:

$$\pi(t) = \nu e^{\mathbf{Q}t} = \nu e^{\mathbf{Q}'(\lambda t)}$$

with $Q' = Q\frac{1}{\lambda}$ and $\lambda = \max_{l\in L} -\mathbf{Q}(l,l)$. That is, we can rescale a (uniform) CTMC with rate λ to a transition rate of 1, by stretching time accordingly. The same holds for time-bounded reachability, as it can be expressed via the transient probability [BKH99]. Below we discuss that the same idea can be exploited to carry over our method on normed Markov games to the full class Markov games.

Markov Games For a Markov game $\mathcal{M} = (L, L_r, L_s, \Sigma, \mathbf{R}, \mathbf{P}, \nu)$ with uniform transition rate $\lambda > 0$, we denote by $\mathcal{M}^{\|\cdot\|} = (L, L_r, L_s, \Sigma, \mathbf{P}, \mathbf{P}, \nu)$ the Markov game that differs from $\mathcal{M}$ only in the rate matrix. (Note that $\mathbf{R} = \lambda\mathbf{P}$.)

It is easy to define a bijection $b : \mathcal{S}[\mathcal{M}^{\|\cdot\|}] \to \mathcal{S}[\mathcal{M}]$ between schedulers of $\mathcal{M}^{\|\cdot\|}$ and $\mathcal{M}$ by mapping a scheduler $s \in S[\mathcal{M}^{\|\cdot\|}, T]$ to the scheduler $s' \in \mathcal{S}[\mathcal{M}, \lambda T]$ with $s'(\mathcal{E}) = s(\lambda\mathcal{E})$ for all extended paths $\mathcal{E} \in \mathit{Paths}(\mathcal{M}) \times \mathbb{R}_{\geqslant 0}$, where $\lambda\mathcal{E}$ multiplies all occurring times in $\mathcal{E}$ by λ.

The time-bounded reachability probability for time bound λT for $\mathcal{M}$ under $s' = b(s)$ is then obviously equivalent to the time-bounded reachability probability for time bound T for $\mathcal{M}^{\|\cdot\|}$ under s. (Colloquially speaking, we stretch time by a simple substitution: We merely change the length of a time unit. Extending the length of a time unit reduces the transition rate—and the time bound.)

Besides lifting the results to general games (uniformisation does not alter the reachability probability), this bijection also provides the recipe for translating

the resulting strategy from the normed game back to the un-normed case: To we simply re-adjust the time unit to its original length.

In the normed games, T is the expected number of discrete transitions. For general games, λT is this expected number, and we can replace all occurrences of T in our Theorems, Lemmata, and Corollaries by λT. The translation of the strategy is equally simple: On a timed history, we take the action that suggested by the history of the normed game for the case that all times (that is, the times where discrete transitions occurred and the current time) were multiplied by λ.

Early schedulers At times [WJ06], schedulers are considered where the action is chosen when a location is entered. It is fairly simple to translate this scheduling problem to one where the scheduler can change its decision over time. The translation essentially encodes the next decision in every location, which is a simple but effective way to encode the making the decision on entry. (See, for example, [RS10] for details.) It is, however, not hard to see that a specialised approach would avoid this minor blow-up. We do not discuss this in detail because we consider the restriction to chose the action on entry as rather artificial.

5 Example

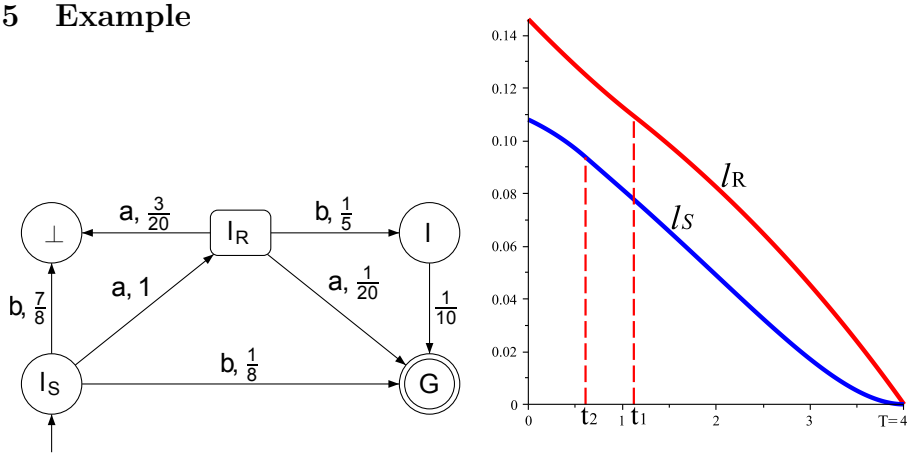

Fig. 3. Left: a normed Markov game, Right: reachability within $[0, 4]$ for l_R and l_S.

We consider the simple normed Markov game in Figure 3, which we use as our running example in the following subsections to exemplify how the ε-nets of the different levels work.

The self-loops of the normed Markov game are not depicted, but as the game is normed, the transition rates for all locations and all enabled control actions are 1; the missing part is assigned to the respective self-loop. l_R is owned by the reachability player, while l_S is owned by the safety player. G and $\bot$ are absorbing, and there is only a single enabled action for l. It therefore does not

matter to which player l, G, and $\bot$ belong. (We have depicted them as vertices of the safety player.) One can assume that the absorbing states have a non-depicted outgoing edge (say, with control action a) to themselves with transition rate 1. In our example, we assume $T = 4$.

The example is constructed such that the analytical solutions can be obtained, which are depicted on the right part in Figure 3 for player l_R and l_S respectively. The graph roughly reflects the development of the time-bounded reachability for the individual locations and points in time for the time bound $T = 4$. Close to $T = 4$, the optimal strategy of the reachability player is to use control action a in l_R and the optimal strategy of the safety player is to use control action a in l_S. There are only two interesting points: at roughly time $t_1 \approx 1.123$ the optimal control action of the reachability player changes from a to b (this happens when the time-bounded reachability at location l reaches 0.25), and around the time 0.609 the optimal control action of the safety player changes to b as well (when the time-bounded reachability at location l_R reaches 0.125).

For our examples, we use $\varepsilon = 0.1$, and focus on the mesh $[1.1, 1.2]$ with initial values $p_k(G, 1.2) = 1$, $p_k(l, 1.2) = 0.244$, $p_k(l_R, 1.2) = 0.107$, $p_k(l_S, 1.2) = 0.075$, $p_k(\bot, 1.2) = 0$, where p_k is the estimator for nets of level k. The values are not the 'true' values for these points, but close enough. we chose them because being more precise with the starting values does not help, as more digits would only obscure the result. We chose the ε mesh in order to make sure that something does happen within the mesh. (Note, however, that the optimal strategies are simply constant in most meshes.)

For our examples, $p_k(G, \cdot) = 1$ and $p_k(\bot, \cdot) = 0$ are constant functions, and we do not mention them on any level.

5.1 Single ε-Net

In single ε-nets, we greedily look at the maximal gain at time $t = 1.2$, and chose the descent to the functions accordingly. The maximising action $a^t_{l_R}$ is a:

$\sum_{l' \in L} \mathbf{R}(l_R, a, l') p_1(l', t) = \frac{1}{20} + \frac{4}{5} \cdot 0.107 > \sum_{l' \in L} \mathbf{R}(l_R, b, l') p_1(l', t) = \frac{1}{5} \cdot 0.244 + \frac{4}{5} \cdot 0.107$.

The minimising action $a^t_{l_S}$ is also a: $\sum_{l' \in L} \mathbf{R}(l_S, a, l') g_{L_r}(l', t) = 0.107 < \sum_{l' \in L} \mathbf{R}(l_S, b, l') g_{L_r}(l', t) = \frac{1}{8}$.

As a result, we obtain the following estimators:

- $p_1(l, t - \tau) = 0.0756\tau + 0.244$,
- $p_1(l_R, t - \tau) = 0.0286\tau + 0.107$, and
- $p_1(l_S, t - \tau) = 0.032\tau + 0.075$.

5.2 Double ε-Net

The simplest case is again location l: There is no choice to make, and we obtain:

- $-\dot{p}_2(l, t - \tau) = 0.1 \cdot (1 - (0.0756\tau + 0.244)) = 0.0756 - 0.00756\tau$ and

- $p_2(l,t) = 0.244$, and hence
- $p_2(l,t-\tau) = -0.00378\tau^2 + 0.0756\tau + 0.244$.

This is the next step of the Taylor expansion of the precise function. For $k > 2$, the estimation p_k can be easily obtained by: $p_k(l,t-\tau) = 0.244 + \sum_{i=1}^{k} \frac{(-0.0756\tau)^i}{i!}$.

Our running example is chosen to ensure that something does happen, namely that the optimal control decision for the reachability player changes. In l_R, we have the situation that

$$\sum_{l' \in L} \mathbf{R}(l_R, a, l') p_1(l', t-\tau) = \frac{1}{20} + \frac{4}{5} p_1(l_R, t-\tau)$$

and

$$\sum_{l' \in L} \mathbf{R}(l_R, b, l') p_1(l', t-\tau) = \frac{1}{5} p_1(l, t-\tau) + \frac{4}{5} p_1(l_R, t-\tau)$$

intersect at $p_1(l, t-\tau) = 0.25$, which happens at $\tau = \frac{60}{756} = \frac{5}{63} := z_2 < \varepsilon = 0.1$ within the ε-mesh. Consequently, we get

- $-\dot{p}_2(l_R, t-\tau) = 0.05 - 0.2(0.0286\tau + 0.107) = -0.00572\tau + 0.0286$ for $0 \leq \tau \leq z_2$ (achieved with optimising action a),
- $-\dot{p}_2(l_R, t-\tau) = 0.2\big((0.0756\tau + 0.244) - (0.0286\tau + 0.107)\big) = 0.0094\tau + 0.0274$ for $z_2 < \tau \leq 0.1$ (achieved with optimising action b),
- $p_2(l_R, t) = 0.107$, and hence
- $p_2(l_R, t-\tau) = -0.00286\tau^2 + 0.0286\tau + 0.107$ for $\tau \leq z_2$ and
- $p_2(l_R, t-\tau) = 0.0047\tau^2 + 0.0274\tau + 0.107047619$ for $\tau > z_2$.

For l_S, the minimising action remains stable, and we obtain $p_2(l_S, t-\tau) = -0.0017\tau^2 + 0.032\tau + 0.075$.

5.3 Triple ε-Net

In l_R, the switching points satisfy, similar to the double ε-Net, the equation $p_2(l, t-\tau) = 0.25$, which happens at $z_3 \approx 0.07968254476$ and $z_3' \approx 19.92031746$: only z_3 is within the ε-mesh. Note $z_3 > z_2 \approx 0.07936507937$. Consequently, we get

- $-\dot{p}_3(l_R, t-\tau) = 0.000572\tau^2 - 0.00572\tau + 0.0286$ for $0 \leq \tau \leq z_2$ (achieved with optimising action a),
- $-\dot{p}_3(l_R, t-\tau) = 0.00094\tau^2 - 0.00548\tau + 0.02859047619$ for $z_2 < \tau \leq z_3$ (achieved with optimising action a),
- $-\dot{p}_3(l_R, t-\tau) = -0.001696\tau^2 + 0.0094\tau + 0.0274$ for $z_3 < \tau \leq 0.1$ (achieved with optimising action b),
- $p_3(l_R, t) = 0.107$, and hence
- $p_3(l_R, t-\tau) = 0.0001906666667\tau^3 - 0.00286\tau^2 + 0.0286\tau + 0.107$ for $0 \leq \tau \leq z_2$ (achieved with optimising action a),

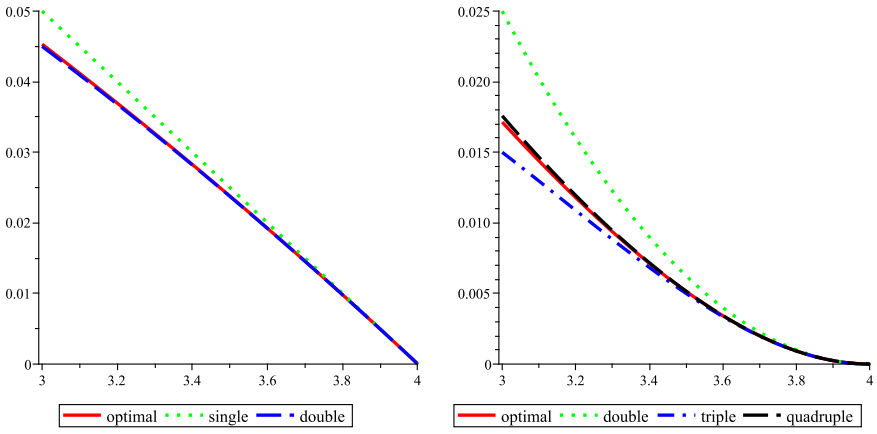

Fig. 4. We consider mesh $[T-1, T]$ with $T = 4$. The initial value for G is one and 0 otherwise. The left part corresponds to the reachability player l_R, and we compare the analytical curve with the estimation obtained by single and double-nets. The estimation under double-nets is already very close. On the right part we have the single, double, triple and quadruple nets estimations for the safety player l_S: while the linear function is constant 0, the estimation under quadruple nets already provides very promising estimations.

- $p_3(l_R, t-\tau) = -0.0003133333333\tau^3 - 0.00274\tau^2 + 0.0286\tau + 0.107$ for $z_2 < \tau \leq t_3$ (achieved with optimising action a),
- $p_3(l_R, t_2-\tau) = -0.0005653\tau^3 + 0.00482\tau^2 + 0.02739047619\tau + 0.1070477458$ for $z_3 < \tau \leq 0.1$ (achieved with optimising action b).

For location l_S, the minimising action remains still stable: but we obtain a piecewise cubic function $p_3(l_S, t-\tau)$ with the switching point z_2 inherited from double nets, as for the reachability player.

5.4 Mesh $[T-1, T]$

For the mesh $[1.1, 1.2]$ discussed in the previous subsections, the estimator p_3 for the triple net is very close to the analytical curve ($\leq 0.510^{-6}$). To illustrate how estimations p_1, p_2, p_3, p_4 converge to the analytical curve, for $T = 4$, we consider the complete interval $[3, 4]$ as a single mesh of maximal length $\varepsilon = 1$.

Note that our techniques are *not* designed for this. The error margins we guarantee are, for a step of length $\varepsilon = 1$, $\mathcal{E}(1,1) = \frac{1}{2}$, $\mathcal{E}(2,1) = \frac{1}{3}$, $\mathcal{E}(3,1) = \frac{1}{6}$, and $\mathcal{E}(4,1) = \frac{1}{15}$. It is, however, necessary to use such big steps to see the difference in precision on a graph.

In Figure 4 shows the resulting estimators for the locations l_R and l_S on the large mesh $[3, 4]$ for nets of different levels.

6 Conclusion

We conclude our paper by comparing the quality of the estimator for single ε-meshes with the state-of-the-art [NZ10] for CTMDPs.

We compare the step length and number of steps for looking for a six, eight, and ten digit precision ($\pi = 5 \cdot 10^{-7}, 5 \cdot 10^{-9}, 5 \cdot 10^{-11}$, respectively) and the inferred mesh length and number of steps for bounded safety or reachability problem with an expected value of ten transitions.

Calculating the required length of a mesh for these precisions, we get

precision	$5 \cdot 10^{-7}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-11}$
state-of-the-art [NZ10]	$5 \cdot 10^{-8}$	$5 \cdot 10^{-10}$	$5 \cdot 10^{-12}$
single nets	10^{-7}	10^{-9}	10^{-11}
double nets	$3.87 \cdot 10^{-4}$	$3.87 \cdot 10^{-5}$	$3.87 \cdot 10^{-6}$
triple nets	$6.69 \cdot 10^{-3}$	$1.44 \cdot 10^{-3}$	$3.11 \cdot 10^{-4}$
quadruple nets	$2.94 \cdot 10^{-2}$	$9.31 \cdot 10^{-3}$	$2.94 \cdot 10^{-3}$

Calculating the resulting numbers of iterations, we get

precision	$5 \cdot 10^{-7}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-11}$
state-of-the-art	200.000.000	20.000.000.000	2.000.000.000.000
single nets	100.000.000	10.000.000.000	1.000.000.000.000
double nets	25.820	258.199	2.581.989
triple nets	1.493	6.934	32.183
quadruple nets	340	1.075	3.399

In each case, the advantage of single nets over the state-of-the-art [NZ10] is negligible, and most likely just a sign that our estimation is marginally better. This is as expected, as the lowest level of our nets was supposed to be the bridge to traditional approaches. The advantage of double nets is enormous, in particular when taking the marginal raise in step-costs (as compared to single nets or state-of-the-art [NZ10] techniques) into account: Compared to single nets the number of steps is reduced by a factor of $\sqrt{1\frac{1}{3} \cdot \frac{T}{\pi}}$.

The advantage of triple nets over double nets is equally impressive on first glance. Yet, we do have to take into account that the individual steps *might* become more expensive. Depending on the parameters of the Markov game in question, this may lead to a worse bound on the cost. The estimations for this are, however, extremely coarse, and we do think that the extra cost per step would not outweigh the savings: a factor of $\sqrt[6]{1\frac{1}{3} \cdot \frac{T}{\pi}}$, in our example a factor between 17.3 and 80.2.

At the very least, it will be worthwhile to consider local values for the number of changes, and use triple nets if the overhead will not be large. The overhead for this is marginal: we only have to evaluate double nets up to the value length of a mesh in triple nets to know how costly it will be to evaluate triple nets. (This does not alter the worst-case cost of the evaluation of this step for the underlying single and double nets, and has little influence on the actual cost. All that can happen is that we have to consider a few more changes in the partially linear function, and could exclude less values values up-front in the improvement described in Subsection 3.2.) In the rare cases where this is too expensive, we

can move forward to evaluating double nets on the respective shorter horizon instead, thus getting the benefits of triple nets without having to take any risk to speak of.

A similar argument can be made for the move from triple to quadruple nets. One should, however, note that the advantage is much smaller: a factor of $\sqrt[12]{5\frac{5}{24} \cdot \frac{T}{\pi}}$, 4.4 to 9.5 in our example. This method may still have its place on the evaluation of parts of the time, namely on those parts where very little happens, but only if the required precision is very high.

An implementation of these techniques is still outstanding, but we expect the practical reduction to be in the region of the reduction of triple nets, as we expect the times where we would rather turn to double nets to be rare. For the expected step costs, we doubt that they will be a full order of magnitude over the step costs for single nets.

References

[BCR+09] Marco Bozzano, Alessandro Cimatti, Marco Roveri, Joost-Pieter Katoen, Viet Yen Nguyen, and Thomas Noll. Verification and performance evaluation of AADL models. In *ESEC/SIGSOFT FSE*, pages 285–286, 2009.

[Bel57] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[BF09] P. Bouyer and V. Forejt. Reachability in stochastic timed games. In *36th International Colloquium on Automata, Languages and Programming (ICALP), Part II*, volume 5556 of *Lecture Notes in Computer Science*, pages 103–114. Springer, 2009.

[BFK+09] Tomás Brázdil, Vojtech Forejt, Jan Krcál, Jan Kretínský, and Antonín Kucera. Continuous-time stochastic games with time-bounded reachability. In *FSTTCS*, pages 61–72, 2009.

[BHKH05] Christel Baier, Holger Hermanns, Joost-Pieter Katoen, and Boudewijn R. Haverkort. Efficient Computation of Time-bounded Reachability Probabilities in Uniform Continuous-time Markov Decision Processes. *Theoretical Computer Science*, 345(1):2–26, 2005.

[BKH99] C. Baier, J.-P. Katoen, and H. Hermanns. Approximate Symbolic Model Checking of Continuous-Time Markov Chains. In *Proceedings of CONCUR'99*, volume 1664 of *Lecture Notes in Computer Science*, pages 146–161, 1999.

[BS11] P. Buchholz and I. Schulz. Numerical analysis of continuous time markov decision processes over finite horizons. *Computers and Operations Research*, 2011.

[CHKM10] Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. Computing maximum reachability probabilities in Markovian timed automata. Technical report, RWTH Aachen, 2010.

[CHLS09] Nicolas Coste, Holger Hermanns, Etienne Lantreibecq, and Wendelin Serwe. Towards performance prediction of compositional models in industrial gals designs. In *CAV*, pages 204–218, 2009.

[GHLPR06] X. P. Guo, O. Hernández-Lerma, and T. Prieto-Rumeau. A survey of recent results on continuous-time Markov decision processes. *TOP*, 14:177–261, 2006.

[GMLS07] Hubert Garavel, Radu Mateescu, Frédéric Lang, and Wendelin Serwe. CADP 2006: A toolbox for the construction and analysis of distributed processes. In *CAV*, pages 158–163, 2007.

[HMW09] Thomas A. Henzinger, Maria Mateescu, and Verena Wolf. Sliding window abstraction for infinite markov chains. In *CAV*, pages 337–352, 2009.

[Mil68a] B. L. Miller. Finite state continuous time Markov decision processes with an infinite planning horizon. *Journal of Mathematical Analysis and Applications*, 22:552–569, 1968.

[Mil68b] Bruce L. Miller. Finite State Continuous Time Markov Decision Processes with a Finite Planning Horizon. *SIAM Journal on Control*, 6(2):266–280, 1968.

[ML67] Anders Martin-Löfs. Optimal control of a continuous-time markov chain with periodic transition probabilities. *Operations Research*, 15:872–881, 1967.

[NSK09] Martin R. Neuhäußer, Mariëlle Stoelinga, and Joost-Pieter Katoen. Delayed Nondeterminism in Continuous-Time Markov Decision Processes. In *Proceedings of FOSSACS '09*, pages 364–379, 2009.

[NZ10] Martin R. Neuhäußer and Lijun Zhang. Time-Bounded Reachability Probabilities in Continuous-Time Markov Decision Processes. In *Proceedings of QEST*, 2010.

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, April 1994.

[RS10] Markus Rabe and Sven Schewe. Finite Optimal Control for Time-Bounded Reachability in CTMDPs and Continuous-Time Markov Games. *CoRR*, abs/1004.4005, 2010.

[Ste94] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton Univ. Pr., 1994.

[WJ06] Nicolás Wolovick and Sven Johr. A Characterization of Meaningful Schedulers for Continuous-Time Markov Decision Processes. In *Proceedings of FORMATS'06*, pages 352–367, 2006.

[ZN10] Lijun Zhang and Martin R. Neuhäußer. Model Checking Interactive Markov Chains. In *Proceedings of TACAS*, pages 53–68, 2010.